SurfLogger: A Logging Browser and Data Processing Method

In Web-based Studies

Jibo He

(Department of Psychology, University of Illinois, Urbana-Champaign, IL, 61820, USA)

Jibohe2@uiuc.edu

ABSTRACT

Although web-based studies are becoming increasingly popular, there are very limited choices of tools available for collecting web-user interactions data. These tools are either highly constrained in the types of data they could collect or limited in availability for the study environment. This paper describes an automated data logging tool, called SurfLogger, for web-based studies. SurfLogger is free, open-source, and easy to modify. SurfLogger is expected to meet the increasing needs of data collection tools for web-based studies.

Keywords

SurfLogger, browser, Web, Python, web-based study

INTRODUCTION

The World Wide Web (WWW) is the fastest developing information resources in this information age (Eighmey, & McCord, 1998). The booming and infinite opportunities accompanying WWW have attracted interests from a vast number of communities, including web site designers, user interface researcher, cognitive psychologists, E-commerce businessman, as well as many others who are interested in characterizing how users interact with web browser (Eighmey, & McCord, 1998; Wang, Jing, He & Yang, 2007; Reeder, Pirolli, & Card, 2000).

Despite the interests in web-based researches, there are still no full-fledged and easily accessible tools to collect user-browser interaction data. Current data collection methods are far from convenient. Some tools collected data from servers or proxies, which are not only bothersome and expensive to configure, but also fail to capture users’ interaction with the browser (Pitkow, 1998). Another substitutive solution is to use videotaped data, usually providing more comprehensive users information (Byrne, John, Wehrle, & Crow, 1999). But coding videotaped data is too consuming in time and labor, and accuracies of coding cannot be guaranteed to be perfect. Reeder, Pirolli and Card (2000) created
WebLogger for data collection in web-based studies. Unfortunately, WebLogger was written in Visual Basics and depends on Microsoft’s Internet Explorer 6.0 (IE) (Reeder, Pirolli & Card, 2000, 2001). WebLogger is often found to be unusable after IE is updated to a new version (7.0).

To meet the demand for such a tool in web-based research, I extended the work in wxPython community, and developed a new tool, called SurfLogger. SurfLogger collects users’ interaction data with the web and the browser.

**SURFLOGGER**

**Description**

SurfLogger (See Figure 1. in Appendix), is written in Python, a scripting language, and the graphical user interface) is created with wxPython (a Python bundle of wxWidget). SurfLogger can record a variety of user actions with the web pages and the browser. SurfLogger produces two files, logfile.txt and urlfile.txt. Logfile.txt stores action IDs (natural numbers assigned to each action, used to track the record to the responding actions), the time stamp for each actions, interaction with the browsers (such as, clicking on the function buttons, such as, Back, Forward, Home, etc.), and coordinates of mouse when clicked. The time stamps could be used to compute the time of completion for each task. The number of button presses on the browsers could be used as a measure of effort in carrying out the task. SurfLogger also captures the images of each screen when the web page refreshes. Drawing the mouse coordinates onto the screen captures could tell us which links the users clicked. Urlfile.txt stores action IDs and URLs (Uniform Resource Locator). Action IDs are used to synchronize the record in logfile.txt and urlfile.txt. URL record is stored in a separate file for the simplicity of offline text processing to extract information imbedded in URLs. Data processing methods are briefly introduced in the section of case study.

SurfLogger is also capable of connecting to external software to record the whole process of user actions. Currently, I use Michael Urman’s Screen Recorder named cankiri as the external software for recording, because it is also written in python and shares the same spirit of open source. With video record, the researchers could know more information about users’ actions. If quality of recording is emphasized, SurfLogger could easily switch to other external recording software, and only one line of the source code (the path of the external software) needs to be changed.

**Advantages and Disadvantages**

Created with scripting language of Python, SurfLogger inherits many advantages of Python. SurfLogger is an automated data logging tool. It is free, open-source, and easy to modify. SurfLogger does not depend on other software, such as Internet Explorer.
SurfLogger also has some limitations. Because Python is slower than C language, SurfLogger usually takes a couple of seconds to startup. SurfLogger needs Python interpreter and several libraries of Python, which may be complicated for Windows users without programming skills. This could be solved by bundling up all required Python libraries and SurfLogger source codes into executable program. There are many free tools for Python to build an executable version of SurfLogger, such as the python packages of py2exe or pyinstaller etc.. The converted executable version of SurfLogger does not need installation of Python in the operating system, and can be run directly, without installation of SurfLogger as well.

Log File Format

The records are stored in two files, the logfile.txt and urlfile.txt. Each variable takes up one line, beginning with the variable name, and followed by the variable value. The variable is named to be self-explanatory. In the logfile.txt (see Figure 1), a record set for one action includes the mouse coordination, browser action (clicking on Back, Forward, Home or other buttons on the browser), time for the action, and action ID. In the urlfile.txt (see Figure 2), each set of record contains action ID and URL. Adjacent record sets are separated by a blank line. The format of log file is designed to be human-readable and easily read for analysis software.

ID: 3
TIME: 04 Apr 2008 11:50:04
Mouse Coordinate: 125 52
Browser Action: Back

Figure 1. Sample records in logfile.txt

ID: 3
URL: http://www.citeulike.org/user/testMaterial/article/2624476

Figure 2. Sample records in urlfile.txt

Case study

To demonstrate how SurfLogger could contribute to web-based research, I will briefly explain the usability analysis of IGroup as a case study (Wang, Jing, He, & Yang, 2007). IGroup is an image search engine, presenting the results in semantic clusters. To test whether IGroup can increase search efficiency compared to MSN, we developed the predecessor of SurfLogger, which functioned similarly like SurfLogger, but less flexible. We developed a measure of Search Effort to compare IGroup and MSN objectively.
Search Effort was defined as the number of query input, and number of links and cluster names clicked by the users. Query input, links and cluster names clicked were extracted from URLs recorded by our automated logging tool. A sample URL recorded in this study is listed as follows:

_Wednesday, August 30, 2006 3:06:54 PM_

http://msra-vss50-b/igroup2/search.aspx?q=Disney#g.14.1,-1

The characters in bold, “Disney”, “14”, and “1” were the input query, ID of cluster name, and result page. The information could be extracted from the URL by simple text processing, using regular expressions. For code of data reduction, URL extraction and source code of SurfLogger please refer to the project page of SurfLogger.

**RELATED WORK**

Although a large number of researchers are interested in web-based studies, there are not many well-developed tools. Reeder, Piroli and Card (2000) developed a the tool called WebLogger, which can collect extensive data, including user input from keyboard and mouse, user actions on the interface elements of IE, and URLs. Choo, Detlor and Turnbull (1999) also developed a similar tool named WebTracker. But both WebLogger and WebTracker can be used only in Windows platform, and relied on the explorer software of IE or Netscape’s Navigator. After the explorers are upgraded, the codes of WebLogger and WebTracker have to be updated in order to function normally.

The LogSquare, sold by ManGold Inc., can record keyboard entries, web page actions, mouse clicks, user comments and coding etc. However, despite the price of LogSquare, it can not offer researchers the flexibilities in data collection and analysis. IT companies also wrote some tools for their usability test. But these tools are usually not full-fledged, and not available to the common researchers (Wang, Jing, He, & Yang, 2007).

Besides the above mentioned automated logging tools, researchers also used some compensatory recording methods. Catledge and Pitkow (1995) studied user interfaces by capturing client-side browsing events in NCSA’s XMosaic. Byrne and his colleague (1999) used videotape recording to study web-browsing behaviors. However, these methods are not only time-consuming, but also provided limited data about users’ behaviors.

**CONCLUSION**

SurfLogger is a useful tool for collecting data for web-based researches. With its many features of automated data logging, and the fact that it is free, open-source, and independent of specific browsers, SurfLogger can be useful for many researchers in data
collection. SurfLogger is expected to contribute more to the increasing interests in web-based researches.

REFERENCES


SurfLogger project page: http://sites.google.com/site/hejibo/surflogger.


Appendix:

(Figure 1 shows a sample screenshot of SurfLogger. Users could browse any web pages as other browsers, while log data is collected in the mean time. The toolbar on the left upper-most lists all the function buttons, Open, Home, Back, etc. The web page on display is part of my ongoing project, using SurfLogger to collect log data.)